

U. S. NAVAL  
PERSONNEL RESEARCH ACTIVITY

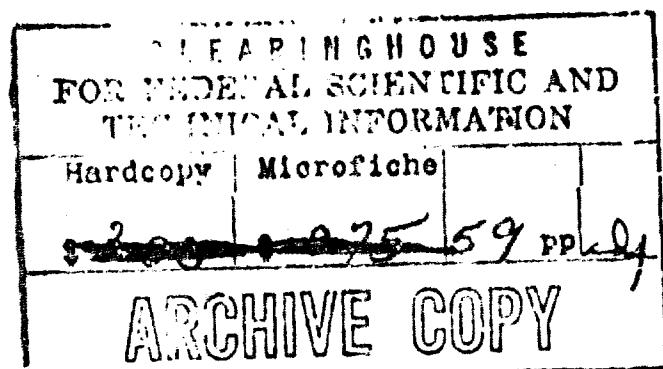
SAN DIEGO, CALIFORNIA 92152

AD62002  
TECHNICAL BULLETIN 65-15

MAY 1965

A COMPUTER PROGRAM FOR THE MAXIMUM  
LIKELIHOOD ANALYSIS OF TYPES

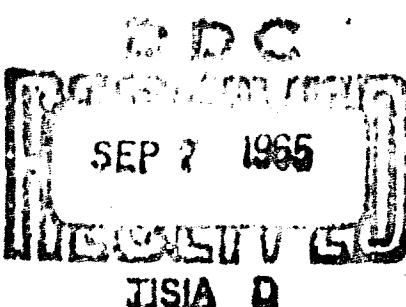
John H. Wolfe



DDC Ref Use only



AN ACTIVITY OF THE BUREAU OF NAVAL PERSONNEL



**Best  
Available  
Copy**

A COMPUTER PROGRAM FOR THE  
MAXIMUM LIKELIHOOD ANALYSIS OF TYPES

JOHN H. WOLFE

U.S. NAVAL PERSONNEL RESEARCH ACTIVITY  
SAN DIEGO, CALIFORNIA 92152

## BRIEF

This report contains a description of a computer program for estimating the parameters of a mixture of multivariate normal distributions with unknown frequencies, means, and covariances. The basic equations for the procedure are presented for the first time here, with their derivation omitted. An example with the results of the computer printout is described for an artificially constructed mixture of three bivariate normal distributions. The method of using the program and the Fortran listing are detailed in this report.

## CONTENTS

BRIEF		PAGE
I	Identification . . . . .	1
II	Purpose. . . . .	1
III	Restrictions . . . . .	1
IV	Method . . . . .	1
V	Usage. . . . .	10
VI	Example - Artificial Clusters. . . . .	12
	A. Listing of Input for Example. . . .	17
	B. Computer Printout for Example . . .	23
VII	Fortran Program Listing . . . . .	30

A COMPUTER PROGRAM FOR THE  
MAXIMUM LIKELIHOOD ANALYSIS OF TYPES

I Identification

A. TYPE

- B. Written by John H. Wolfe, January 1963. Revised June, 1964.  
C. U.S. Naval Personnel Research Activity, San Diego, California.  
D. Coded entirely in FORTRAN II.

II Purpose

Given  $m$  scores on each of  $N$  individuals drawn from an unknown mixture of multivariate normal distributions, the program gives maximum-likelihood estimates of the means and covariances within each type, the relative frequency of each type, and the maximum likelihood of the sample for a given number of types.

III Restrictions

A. The program uses 14,554<sub>8</sub> words in the main program and 47,256<sub>8</sub> words in COMMON. One input magnetic tape (Unit #3), one output magnetic tape (Unit #2) and one tape for temporary binary storage (Unit #4) are required.

B. Restrictions on Parameters

Numbers of Variables  $\leq 5$

Number of Types  $\leq 6$

IV Method

The program solves the maximum-likelihood equations by one of four alternative iteration schemes, depending on a control card option.

Suppose that  $m$  measurements have been made on  $N$  individuals. Let  $x_{ik}$  be the  $i^{\text{th}}$  variable for individual  $k$ . Suppose the population from which the sample of  $N$  individuals is drawn is a mixture of  $r$  multivariate normal distributions. That is, the probability density,  $f(\bar{x})$ , is given by

$$f(x) = \sum_{s=1}^r \lambda_s a_s(x), \text{ where } \sum_{s=1}^r \lambda_s = 1, \lambda_s > 0, \text{ and}$$

$$a_s(x) = \left( \frac{\sigma_s^{ij}}{(2\pi)^m} \right)^{1/2} e^{-1/2 \sum_{ij} (x_i - M_i^s)^2 / \sigma_s^{ij}}$$

Here  $\lambda_s$  is the relative proportion of type  $s$  in population,  $|\sigma_s^{ij}|$  is the determinant of the inverse of the covariance matrix for type  $s$ , and  $M_i^s$  is the mean of the  $i^{\text{th}}$  variable for type  $s$ .

Let us define  $g_s(x) = \alpha_s(x) / f(x)$ . SUBROUTINE DENSITY calculates  $\alpha_s$ ,  $f$ , and  $g_s$  for each individual. Define the "generalized sample moments"  $\{\mu_{ijab}^{ps}\}$  as follows:

$$\mu_{ijab}^{ps} = \frac{1}{N} \sum_{k=1}^N x_{ik} x_{jk} x_{ak} x_{bk} g_s(x_k) g_p(x_k)$$

If a subscript is omitted or set to 0, the corresponding term on the right hand side of the equation is omitted. For example

$$\mu_{ijao}^{po} = \mu_{ija}^p = \frac{1}{N} \sum_{k=1}^N x_{ik} x_{jk} x_{ak} g_p(x_k)$$

$$\text{and } \mu_0^o = \frac{1}{N} \sum_{k=1}^N 1 = 1.$$

SUBROUTINE MOMENT computes a table of  $\{\mu_{ijab}^{ps}\}$  with  $b \leq a \leq i$  and  $p \leq s$ .

FUNCTION U (II, IJ, IA, IB, IP, IS) looks up the correct  $\mu_{ijab}^{ps}$  from the two-dimensional table created by MOMENT, even when the inequalities on the indices are not satisfied.

The maximum likelihood estimates of the parameters are those values which maximize the function

$$L = \sum_{k=1}^N \log f(x_k) - \omega \left( \sum_{s=1}^r \lambda_s - 1 \right).$$

Setting the partial derivatives of the likelihood to zero results in the following equations:

$$f_{oo}^s = \mu_0^s - 1 = 0$$

$$f_{oi}^s = \mu_i^s - \mu_o^s M_i^s = 0$$

$$f_{ij}^s = \mu_{ij}^s - M_i^s \mu_j^s - M_j^s \mu_i^s - \mu_o^s (\sigma_{ij}^s - M_i^s M_j^s) = 0$$

The moments  $\{\mu_{ij}^s\}$  can also be differentiated as follows:

$$\frac{\partial \mu_{ij}^s}{\partial \lambda_p} = -\mu_{ij}^{ps}$$

$$\frac{\partial \mu_{ij}^s}{\partial M_a^p} = -\lambda_p \sum_{b=1}^m \sigma_p^{ab} (\mu_{ijb}^{ps} - M_b^p \mu_{ij}^{ps}) + \delta_{ps} \sum_{b=1}^m \sigma_p^{ab} (\mu_{ijb}^p - M_b^p \mu_{ij}^p)$$

$$\frac{\partial \mu_{ij}^s}{\partial \sigma_p^{ab}} = (1 - \frac{\delta_{ab}}{2}) \{-\lambda_p [\mu_{ij}^{ps} (\sigma_{ab}^p - M_a^p M_b^p) + M_a^p \mu_{ijb}^{ps} + M_b^p \mu_{ija}^{ps} - \mu_{ijab}^{ps}]$$

$$+ \delta_{ps} [\mu_{ij}^p (\sigma_{ab}^p - M_a^p M_b^p) + M_a^p \mu_{ijb}^p + M_b^p \mu_{ija}^p - \mu_{ijab}^p]\}$$

Where  $\delta_{ps} = 1$  if  $p=s$   
0 otherwise

The derivatives of the moments are computed by

$$\text{FUNCTION DERV (II, IJ, IS, IA, IB, IP)} = \frac{\partial \mu_{ij}}{\partial \theta_{ab}^p},$$

Where  $\theta_{oa}^p = M_a^p$ ,  $\theta_{oo}^p = \lambda_p$ , and  $\theta_{ab}^p = \sigma_p^{ab}$  for  $a \neq b$ .

DERV calls on three functions, ONE, EM, SIG defined as follows:

$$\text{ONE (IA, IB)} = \delta_{ab}$$

$$\text{EM ( II, IJ, IA, IS, IP)} = \sum_{b=1}^m \sigma_p^{ab} (\mu_{ijb}^{ps} - M_b^p \mu_{ij}^{ps})$$

$$\text{SIG (II, IJ, IA, IB, IS, IP)} = u_{ij}^{ps} (\sigma_{ab}^p - M_a^p M_b^p) + M_a^p u_{ijb}^{ps}$$

$$+ M_b^p u_{ija}^{ps} - u_{ijab}^{ps}$$

The function EM used with  $s=0$  gives the second term in  $\frac{\partial u_{ij}^s}{\partial M_a^p}$

and SIG used with  $s=0$  gives the second term in  $\frac{\partial u_{ij}^s}{\partial \sigma_{ab}^p}$ .

The maximum likelihood equations  $\{f_{ij}^s = 0\}$  can be solved in several ways. One iterative scheme is

Newton-Raphson iteration, which solves the linear equations

$$\sum_{p=1}^r \sum_{b=1}^m \sum_{a=b}^m \frac{\partial f_{ij}^s}{\partial \theta_{ab}^p} \Delta \theta_{ab}^p = -f_{ij}^s$$

for  $\Delta \theta_{ab}^p$ .

On the next iteration  $\theta_{ab}^p = \theta_{ab}^p + \Delta \theta_{ab}^p$ .

SUBROUTINE NEWTON computes the vector

$B(IAT) = -f_{ij}^s$  and the matrix of coefficients

$A(IAT, JAT) = \partial f_{ij}^s / \partial \theta_{ab}^p$ .

SUBROUTINE MATINV, a standard SHARE routine, solves the linear equations for  $\Delta \theta_{ab}^p$  and stores the result in the vector B (IAT).

SUBROUTINE RAPISON computes the values of the parameters for the next iteration. First it determines if any of the increments are so large that  $\lambda_p + \Delta \lambda_p < 0$  or  $> 1$ . If so, the increment vector B is shortened until no  $\Delta \lambda_p$  moves  $\lambda_p$  more than half the interval between  $\lambda_p$  and a boundary point.

That is, if  $\Delta \lambda_p < 0$ ,  $\Delta \lambda_p > -\frac{\lambda_p}{2}$ , and if  $\Delta \lambda_p > 0$ ,  $\Delta \lambda_p < \frac{1-\lambda_p}{2}$ . After the  $\Delta \theta_{ab}^p$

are shortened, they are added to the old estimates of the parameters to obtain new estimates for the next iterations.

The main routine also plays a role in shortening the increment vector. If the new likelihood is less than the previous one, or if the determinant of one of the covariance matrices as determined by RAPISON is negative, then the increment vector B is shortened to half its previous value and subroutine RAPISON is entered again.

Several alternative versions of  $\{f_{ij}^S\}$  can be written.

At the maximum-likelihood points

$$f_{00}^S = \mu_0^S - 1 = 0, \text{ hence } \mu_0^S = 1.$$

$$f_{0i}^S = \mu_i^S - \mu_0^S M_i^S = \mu_i^S - M_i^S = 0, \text{ hence } \mu_i^S = M_i^S.$$

Substituting for  $\mu_i^S$  and  $\mu_0^S$  in  $f_{ij}^S$ , we have

$$f_{ij}^S = \mu_{ij}^S - M_i^S M_j^S.$$

When METH = 1 on the control card, the subroutine NEWTON iteratively solves the equations

$$1 f_{00}^S = \mu_0^S - 1 = 0.$$

$$1 f_{0i}^S = \mu_i^S - M_i^S = 0.$$

$$1 f_{ij}^S = \mu_{ij}^S - M_i^S M_j^S = 0.$$

When METH = 2, subroutine NEWTON solves the original set of equations, hereafter referred to as  $\{2 f_{ij}^S = 0\}$

When METH = 3, the equations used are

$$3 f_{00}^S = 1/\mu_0^S (2 f_{00}^S) = 1 - 1/\mu_0^S = 0.$$

$$3 f_{0i}^S = 1/\mu_0^S (2 f_{0i}^S) = \mu_i^S/\mu_0^S - M_i^S = 0.$$

$$3 f_{ij}^S = 1/\mu_0^S (2 f_{ij}^S) = \mu_{ij}^S - M_i^S \mu_j^S/\mu_0^S - M_j^S \mu_i^S/\mu_0^S - (c_{ij}^S - M_i^S M_j^S) = 0.$$

All three types of equations have the same solutions but their radii of convergence may differ.

For METH = 2,

the function  $\{{}_2f_{ij}^s\}$  can be summarized in the following formula:

$$\begin{aligned} {}_2f_{ij}^s &= u_{ij}^s - \delta_{io} + (1 - \delta_{io})\{\theta_{ij}^s u_j^s + (1 - \delta_{jo})[u_o^s(\sigma_{ij}^s - \theta_i^s \theta_j^s) \\ &\quad + u_j^s \theta_i^s]\}\}. \end{aligned}$$

The partial derivatives of  $f_{ij}^s$  are then easily written as

$$\begin{aligned} \frac{\partial {}_2f_{ij}^s}{\partial \theta_{ab}^p} &= \frac{\partial u_{ij}^s}{\partial \theta_{ab}^p} - (1 - \delta_{io})\{\theta_j^s \frac{\partial u_j^s}{\partial \theta_{ab}^p} + (1 - \delta_{jo})[\frac{\partial u_o^s}{\partial \theta_{ab}^p} (\sigma_{ij}^s - \theta_j^s \theta_i^s) + \theta_j^s \frac{\partial u_i^s}{\partial \theta_{ab}^p}]\} \\ &\quad - \delta_{ps}(1 - \delta_{io})\{\delta_{bo}[{}_{2b}u_j^s + (1 - \delta_{jo})(\delta_{ja}u_i^s - \delta_{ia}u_o^s \theta_j^s - \delta_{ja}u_o^s \theta_i^s)] \\ &\quad - (1 - \frac{\delta_{ab}}{2})(1 - \delta_{bo})(1 - \delta_{jo})u_o^s (\sigma_{ia}^p \sigma_{jb}^p + \sigma_{ib}^p \sigma_{ja}^p)\} \end{aligned}$$

where  $0 \leq i \leq m$  and  $0 \leq b \leq a \leq m$ .

The above formulas are used in METH = 2. If METH = 1, the functions

$${}_1f_{oo}^s = u_o^s - 1$$

$${}_1f_{oi}^s = u_i^s - M_i^s$$

$${}_1f_{ij}^s = u_{ij}^s - (\sigma_{ij}^s + M_i^s M_j^s)$$

are summarized by the formulas

$${}_1f_{ij}^s = u_{ij}^s - \delta_{io} + (1 - \delta_{io})\{\delta_{jo}\theta_i^s + (1 - \delta_{jo})[\gamma_{ij}^s + \theta_i^s \theta_j^s]\}, \text{ and}$$

$$\begin{aligned} \frac{\partial {}_1f_{ij}^s}{\partial \theta_{ab}^p} &= \frac{\partial u_{ij}^s}{\partial \theta_{ab}^p} - (1 - \delta_{io})\delta_{ps}\{\delta_{bo}\delta_{jo}\delta_{ia} + (1 - \delta_{jo})(\delta_{ia}\theta_j^s + \delta_{ja}\theta_i^s)\} \\ &\quad - (1 - \frac{\delta_{ab}}{2})(1 - \delta_{bo})(1 - \delta_{jo})(\sigma_{ia}^p \sigma_{jb}^p + \gamma_{ib}^p \sigma_{ja}^p)\}. \end{aligned}$$

The functions  $\{_3 f_{ij}^s\} = \{_2 f_{ij}^s / \mu_o^s\}$ .

$$\text{Hence } \frac{\partial _3 f_{ij}^s}{\partial \theta_{ab}^p} = \frac{1}{\mu_o^s} - \frac{\partial _2 f_{ij}^s}{\partial \theta_{ab}^p} = \frac{_2 f_{ij}^s}{(\mu_o^s)^2} - \frac{\partial \mu_o^s}{\partial \theta_{ab}^p}.$$

The maximum likelihood Newton-Raphson iteration equations give:

$$\sum_{abp} \frac{\partial _3 f_{ij}^s}{\partial \theta_{ab}^p} \Delta \theta_{ab}^p = -_3 f_{ij}^s.$$

Multiplying by  $\mu_o^s$  and substituting, we have

$$\sum_{abp} \left[ \frac{\partial _2 f_{ij}^s}{\partial \theta_{ab}^p} - _2 f_{ij}^s \left( \frac{1}{\mu_o^s} - \frac{\partial \mu_o^s}{\partial \theta_{ab}^p} \right) \right] \Delta \theta_{ab}^p = -_2 f_{ij}^s.$$

Thus the equations for METH = 3 are readily calculated from the equations for METH = 2. Only the matrix of coefficients has to be

changed, simply by subtracting  $_2 f_{ij}^s \left( \frac{1}{\mu_o^s} - \frac{\partial \mu_o^s}{\partial \theta_{ab}^p} \right)$ .

Instead of Newton-Raphson iteration, a method of successive substitutions may be used for finding  $M_i^s$  and  $\sigma_{ij}^s$ .

Since  $f_{oi}^s = 0 = \mu_i^s - \mu_o^s M_i^s$ , the value of  $M_i^s$  for the next iteration is defined as  $M_i^s = \mu_i^s / \mu_o^s$ .

Since  $f_{ij}^s = 0 = \mu_{ij}^s - M_i^s \mu_i^s - M_j^s \mu_j^s - \mu_o^s (\sigma_{ij}^s - M_i^s M_j^s)$  we can solve

for  $\sigma_{ij}^s$  after first substituting  $M_i^s \mu_o^s$  for  $\mu_i^s$ :

$\sigma_{ij}^s = \mu_{ij}^s / \mu_o^s - M_i^s M_j^s$ , where  $M_i^s$  are the new values =  $\mu_i^s / \mu_o^s$ .

The new values of  $\lambda_p$  can be determined by Newton-Raphson iteration using only the equations  $\{f_{00}^s = \mu_o^s - 1 = 0\}$

Differentiation of these equations leads to the system

$$\sum_{p=1}^r \frac{\partial \mu_o^s}{\partial \lambda_p} \Delta \lambda_p = -\mu_o^s + 1$$

$$\text{or } \sum_{p=1}^r u_o^{ps} \Delta \lambda_p = u_o^s - 1$$

When the control card METH = 0, subroutine NEWTON determines the increments associated with successive substitutions.

Experience with the program seems to indicate that the NEWTON-Raphson iteration schemes have very small radii of convergence as compared with the successive substitution methods. Theoretically, however, the Newton-Raphson iteration should converge more rapidly once within its radius. Therefore, provision has been made in the program for running a fixed number of iterations with successive substitutions so as to get improved initial estimates and then switching to Newton-Raphson methods for the remaining iterations. If the control card option is -1, -2, or -3, then a certain number of iterations by successive substitution will be used before Newton-Raphson iteration by methods 1, 2, 3 respectively. The control card number IDUMP specifies the number of preliminary iterations.

The subroutine INITIAL determines the initial values of parameters preliminary to iteration. The proper determination of initial values is crucial to the successful convergence of any iteration method. The initial values determined by INITIAL are quite crude, and the researcher may wish to write his own version of INITIAL after some experimentation. The present version also allows the user to specify his own guesses of the initial values of the parameters of certain types.

The subroutine INITIAL takes a sample of 100 individuals and subjects them to a crude clustering procedure. For each individual, a count is made of the number of other individuals within a "box" two standard deviations on a side around it. That is, for individual k, the number of individuals j is counted such that  $|x_{ik} - x_{ij}| \leq c_j$  for all  $i = 1, 2, \dots, m$ . The individual with the highest count is made the centroid of the first cluster--that is, his scores are the initial estimates of the means for the first type.

The individuals in the first cluster are erased from the sample and the procedure is repeated with the remaining individuals.

The initial estimates of the  $\lambda_s$  are all equal to  $1/r$ , where r is the number of types.

The initial estimates of the covariances are the same for each type and are equal to the covariances computed from the sample of 100 taken as a whole.

After each iteration the subroutine RRESULT prints the current estimates of the parameter for each type. At the end of the last iteration, the program PLACE gives the probabilities of membership in each type for each individual.

These probabilities are:

$$P(\text{individual } k \text{ & Type } S) = \lambda_s g_s(x_k).$$

A few words should be said about the indexing used within the program. First of all, the indices of the moments do not range from 0 to  $m$  and 0 to  $r$  as in our equations, but from 1 to  $m+1$  and 1 to  $r+1$ . Thus the value of  $U(2, 3, 1, 1, 1, 4) = u_{12}^3$ . The moments are conveniently calculated by

setting  $z_{i+1} = x_i$  for each  $x_{ik}$

and  $z_1 = 1.0$ . Similarly  $G(1) = 1.0$

and  $G(2)$  is the relative density for type 1,  $g_1(x_k)$ .

The values for PERS (K) =  $\lambda_{k-1}$

$$\text{COV}(I, J, K) = \sigma_{ij}^{k-1} \text{ and COVIN}(I, J, K) = \sigma_{k-1}^{ij}.$$

also  $\text{AV}(I, K) = M_i^{k-1}$ .

The routine MOMENT collapses the  $\{u_{ijab}^p\}$  into a two dimensional array.

The single index KL is uniquely related to p and s and the single index IJ'N is uniquely related to the indices i, j, a, and b.

In general, suppose we have an array indexed as follows:

There are M indices. The first index varies from 1 to N. Each succeeding index varies from 1 to the preceding index.

$$1 \geq IX(1) \geq \dots \geq IX(M)$$

Let  $S(N, M)$  = number of elements in this array.

$$\text{Then } S(N, M) = \sum_{I=1}^N S(I, M-1)$$

$$\text{and } S(N, M) = \frac{(N+M-1)!}{M! (N-1)!} = \binom{N+M-1}{M}.$$

Let  $IX(1) \geq IX(2) \geq \dots \geq IX(M)$  be a sequence of indices for a particular element of the array. The one-dimensional index of the element is

$$K = 1 + \sum_{I=1}^M \sum_{J=1}^I (IX(M-I+1)-2+J)/J$$

$$\text{or } K = 1 + \sum_{I=1}^M S(IX(M-I+1)-1, I).$$

These formulas are used by the Function U to look up values of  $\nu_{ijab}^{ps}$ .

## V Usage

### A. Input(TAPE Unit 3, BCD-card images)

1. Title Card in columns 1-72, any alphanumeric characters.
2. Control Card

COLS:	NAME	DEFINITION
1-4	MX	Number of variables
5-12	NX	Sample size
13-16	IRM	Number of Types Assumed. If IRM = 0, 6 analyses will be done assuming 1, 2, 3, 4, 5, 6 types.
17-20	ITERM	Maximum Number of iterations. If blank, ITERM is set to 50.
21-28	CONV	Criterion of Convergence which all parameters must satisfy between successive iterations. If blank, CONV is set to .0001.
29-32	IRUN	=1 if every iteration is printed, = 0 if only the last iteration is printed
33-36	MTHI	=0 if successive substitutions is used $\pm 1$ , $\pm 2$ , $\pm 3$ , if various Newton-Raphson methods are used.
37-40	IDUMP	The number of preliminary iterations by successive substitutions before Newton-Raphson iteration for MTHI = -1, -2, or -3.

3. Variable Format Card. This is an ordinary FORTRAN Variable Format Card according to which the data will be read.

### 4. Data Deck

N sets of one or more cards per individual.

### 5. Initial Estimates of Parameters(optional)

#### (a) Estimate control card

Cols 1-4 = K = TYPE # (1 through 6)

Cols 5-8 =  $\lambda_k$  = Proportion of population of type K  
(all 4 digits assumed after decimal point)

#### (b) Means for type K

8 digits per mean, last 4 digits assumed after decimal point.

#### (c) Standard deviations for type K (same format as (b))

#### (d) Correlation matrix for type K (1 row per card, same format as (b))

#### (e) Estimate control card for another type, etc.

A blank card terminates the reading of initial estimates. If no initial estimates are to be read, one blank card must be read. The sets of initial estimates do not have to be present for all types. For example, a set of initial estimates for type 5 may be followed by a set for type 3 followed by a blank card. If any initial parameters are read

for a given type, all parameters must be read for that type. For example, initial estimates of means for type 3 must be followed by initial estimates of standard deviations and correlations for type 3.

If no initial estimates of parameters are read in, the computer will generate its own by a clustering procedure.

Multipie runs may be made at one time by placing one batch of input cards [A(1) to A(5)] followed by another batch. The last batch of data must be followed by two blank cards.

B. Recommended typical usage.

Under most conditions, the control card A-2 will be blank in all but the first 12 columns. The composition of the input will be:

Title Card  
Control Card (Cols 1-12 only)  
Variable Format Card  
Data Cards  
3 blank cards.

Another highly successful set of control parameters is METH = -1 or -2 with IDUMP = 40

C. Output (Tape No 2)

The natural logarithm of the likelihood is printed for each

iteration:  $\sum_{k=1}^N \log f(x_k) - N(\sum_{s=1}^r \lambda_s - 1)$ . If the likelihood on an

iteration is not greater than that on all previous iterations, the computer prints out "Iteration--diverges". The parameters for each type are printed out on either the last iteration or on every iteration, depending on the value of IRUN. At the end of the last iteration, the probabilities of type membership for each individual,  $\lambda_s g_s(x_k)$ , are printed. If sense switch 3 is down, the computer dumps out the moments  $\{u_{ijab}^{ps}\}$  and the matrices  $\{\sigma_{ij}^{ss}\}$  and  $\{\sigma_{ij}^{ij}\}$  following subroutine MOMENT, the matrix A of coefficients of the Newton-Raphson iteration and the vector  $\{-f_{ij}^{ss}\}$  following subroutine NEWTON, and the matrices  $A^{-1}$  and the vector  $\{\Delta \theta_{ij}^p\}$  following the subroutine MATINV.

D. Time estimates.

For METH = 0, time per iteration in seconds is

$$T = .14 N (m+1)(m+2)(r+1)$$

For METH = 0, time per iteration in seconds is

$$T = .0075 (m+1)(m+2)(m+3)(m+4)(r+1)(r+2)N$$

$$+ .00002[(m+1)(m+2)r]^3 N$$

A typical run with  $N = 225$ ,  $r = 1, 2, 3, 4, 5, 6$ ,  $m = 2$  required 50 minutes by METH = 0 on the CDC 1604.

E. Suggestions for reducing running time.

1. By intuition, cluster analysis of variables, factor analysis, or any other means, reduce the number of variables to those that are really important for discriminating types.

2. Use any good prior information from theoretical hypotheses or any other classification-clustering procedures to provide initial estimates.

3. If you have a large sample, say 10,000, don't run it all at once. Take a small sample, say 100 to 200 cases at random and run an analysis using a weak convergence criterion, say .01. Get an idea of how many types there are and some initial estimates from the small sample. Run the entire set of data using these initial estimates on specified numbers of types. The hypothesis  $H_r$  that there are r types can be tested against the alternative,  $H_{r-1}$  that there are r-1 types by the  $\chi^2$  test

$$\chi^2_{d.f.} = -2 \log \lambda = 2(L_r - L_{r-1})$$

with d.f. =  $(m+1)(m+2)/2$

where  $L_r$  = likelihood printed for r types. Thus the

$\chi^2$  test of the likelihood ratio on the small sample may give a good guess as to how many types there are. A final test on the entire sample could be performed specifying r, r+1, and r-1 types on the runs, where r is the hypothesized number of types.

VI Example - Artificial Clusters

To test this program, an example was constructed consisting of three artificial clusters in two dimensions. The points in each cluster were generated by a pseudo-random normal deviate generator. The characteristics of each cluster are given in Table 1 below. The results of the computer printout are summarized in Table 2. The points are plotted in Figure 1. The 75 points in Cluster 1 are designated by triangles in Figure 1, the 50 points in Cluster 2 are designated by squares, and the 100 points in Cluster 3 are designated by circles. Drawn around some of the squares, circles, and triangles are larger squares, circles and triangles. The larger symbols give the classification assigned by the computer program with 3 specified types. If a point does not have two symbols around it, it was correctly classified by the computer. It is evident from the figure that most points were correctly classified, and the computer's types have clear cut boundaries whereas the actual clusters overlap to some degree.

TABLE 1  
Characteristics of Artificial Clusters

	Cluster 1		Cluster 2		Cluster 3	
Number/225	.333		.222		.444	
MEANS	.05	-1.33	-.83	1.64	1.41	.85
S.D. 's	1.26	.49	.87	1.04	.92	.97
$r_{12}$	.1590		.4049		.4743	

TABLE 2  
Characteristics of Types from Computer Program

	Type 2		Type 3		Type 1	
$\lambda_p$	.346		.170		.484	
Means	.20	-1.31	-.11	1.79	1.19	.93
S.D. 's	1.28	.50	.83	1.12	1.04	.91
$r_{12}$	.2462		.7169		.5231	

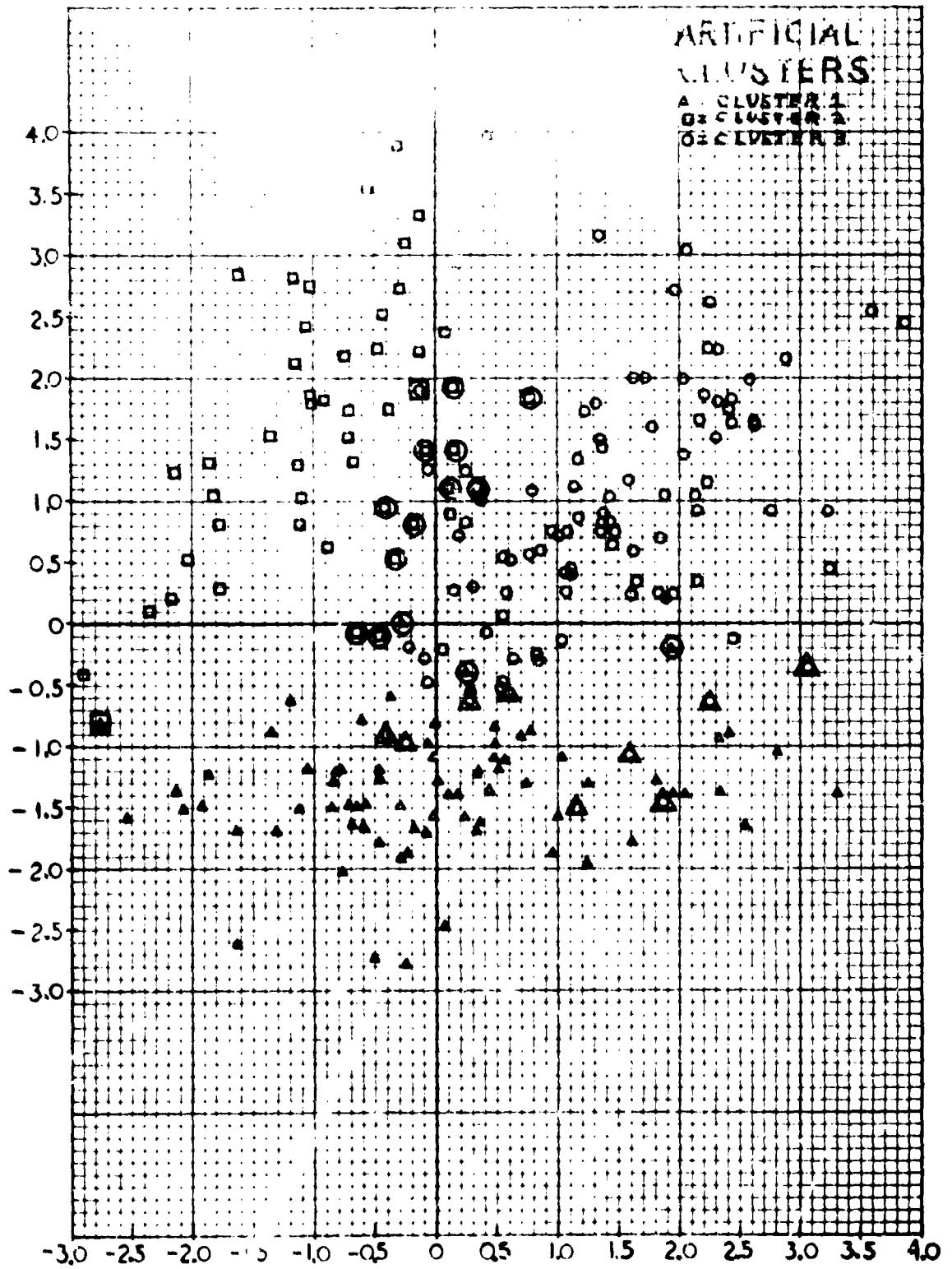


TABLE 3  
Likelihoods and  $\chi^2$  for Numbers of Types

Number of Types	Natural Logarithm of Likelihood	$\chi^2$ (with 6d.f.) $= 2(L_R - L_{R-1})$	P
1	-380.48930		
2	-358.96468	43.04924	.114 $\times 10^{-6}$
3	-340.36400	37.20136	.161 $\times 10^{-5}$
4	-334.83078	11.06644	.863 $\times 10^{-1}$
5	-325.63847	18.38462	.534 $\times 10^{-2}$
6	-318.02872	15.21950	.186 $\times 10^{-1}$

The data cards for input are listed in Section B and the output is given in Section C.

A previous run with METH = 0, IRM = 0 (not given here) gave the likelihoods for 1 to 6 types. These are presented in Table 3, along with associated  $\chi^2$  values. The results indicate that the hypothesis that there are only two types can be rejected against the hypothesis that there are three types; but the hypothesis that there are three types cannot be rejected against the alternative that there are four types.

In order to save space in this report, only a run with IRM = 3 is given in Sections B and C.

The method used was METH = -1, which has 10 preliminary iterations by successive substitutions followed by Newton-Raphson iteration. The previous unpublished run with METH = 0 took 45 iterations to converge, while the one presented here took only 17. The difference can be attributed to the superior convergence rate of Newton-Raphson methods. However, other computer runs with IRM = 0, METH = -1 and IDUMP = 10, 30, or 40 sometimes failed to converge at all, once the Newton-Raphson procedure was started. If the likelihoods have not converged to 0.1 by the successive substitutions, then Newton-Raphson iteration often fails. Thus the initial estimates must be quite accurate if Newton-Raphson iteration is to work. So far the various methods -1, -2, -3, appear to work equally well. All three converged in exactly 17 iterations in the present example. The run reported here took 13 minutes on the CDC 1604.

## **SECTION VI**

### **A. Listing of Input for Example**

ARTIFICIAL CLUSTERS METHOD - 1 PRELIMINARY ITERATIONS

	2	275	3	-1	-6
(2F4.2)					001
-048-141					002
-018-157					003
103-017					004
-057 002					005
-0 2-151					006
-016-165					007
054-157					008
-076-165					009
-177-150					10
0 0-075					011
-2 4-131					012
0 4-243					013
2-1-100					014
272-090					015
-047-173					016
-042-118					017
-241-157					018
2 0-134					019
240-089					020
3 0-138					021
041-125					022
-043-129					023
-247-140					024
1 3-108					025
143-122					026
074-114					027
015-160					028
-164-146					029
-084-148					030
082-068					031
-077-200					032
108-134					033
-136-085					034
-170-168					035
171-191					036
-048-113					037
1 2-129					038
243-161					039
-036-055					040
-080-119					041
-040-162					042
-003-104					043
-072-141					044
-276-077					045
049-107					046
050-115					047
011-139					

-162-257	48
-166-121	49
34-118	50
-164-166	51
24-272	52
77-125	53
43-144	54
21-144	55
49-121	56
-128-115	57
74-246	58
61-073	59
60-270	60
16-125	61
232-133	62
66-146	63
-127-060	64
24-184	65
32-169	66
49-031	67
25-034	68
54-127	69
48-160	70
94-185	71
100-152	72
161-174	73
7-026	74
26-187	75
70 134	76
47 229	77
32 278	78
77 186	79
73 127	80
-107 246	81
-235-012	82
-219 025	83
59 257	84
-284 147	85
-179 081	86
-111 105	87
13 145	88
74 220	89
71 155	90
46 241	91
68-004	92
14 337	93
-115 292	94
-114 215	95
-179 031	96
26 312	97

41	98	98
35	055	99
-103	280	100
-292	-038	101
92	124	102
-205	053	103
-186	108	104
18	084	105
-215	128	106
45	400	107
49	-~	108
-161	288	109
7	143	110
-111	082	111
11	111	112
90	066	113
34	112	114
-113	135	115
13	196	116
11	091	117
-186	130	118
106	240	119
-103	107	120
40	178	121
-138	157	122
11	226	123
32	391	124
-101	185	125
63	-027	126
131	180	127
62	052	128
275	092	129
107	042	130
208	306	131
38	105	132
218	137	133
113	-150	134
139	147	135
27	110	136
14	190	137
60	-054	138
240	179	139
161	200	140
141	084	141
86	061	142
7	129	143
134	151	144
172	200	145
230	193	146
160	025	147

13 030	
17 071	148
214 107	149
135 078	150
224-063	151
204 224	152
103-012	153
9-049	154
56 056	155
181 025	156
163 06^	157
189 106	158
261 163	159
205 139	160
193 126	161
359 255	162
118 187	163
109 074	164
2.1 038	165
23 083	166
322 093	167
11 109	168
54-042	169
111 112	170
10-028	171
112 040	172
106 029	173
211 093	174
148 118	175
23 128	176
6 027	177
136 083	178
79 059	179
2-020	180
31-089	181
139 092	182
287 217	183
44-002	184
241 185	185
179 160	186
96 079	187
58 009	188
221 188	189
232 180	190
222 117	191
69-027	192
168 078	193
135 316	194
197 270	195
25-095	196
	197

203	200	198
80	110	199
175	122	200
233	225	201
245	167	202
261	151	203
23-018		204
189	022	205
218	169	206
226	261	207
96	018	208
259	197	209
31	032	210
112	141	211
385	246	212
306-035		213
242-010		214
142	106	215
198-102		216
107	076	217
24-054		218
144	064	219
324	046	220
82-022		221
163	037	222
51-050		223
182	071	224
175-143		225
		BLANKO <sup>1</sup>
		BLANKO <sup>C</sup>
		BLANKO <sup>U</sup>

**SECTION VI**

**B. Computer Printout for Example**

```

' ITERATION 1, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.42371292E+03
ITERATION 2, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.37195046E+03
ITERATION 3, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.36744158E+03
ITERATION 4, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.364440563F+03
ITERATION 5, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.36004806F+03
ITERATION 6, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.35693192E+03
ITERATION 7, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.35270142F+03
ITERATION 8, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.34901014F+03
ITERATION 9, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.34678251F+03
ITERATION 10, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.34476925F+03
ITERATION 11, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.34381568F+03
ITERATION 12, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.3465752AF+03
ITERATION 11 DIVERGES
ITERATION 13, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.34173161F+03
ITERATION 14, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.34044023F+03
ITERATION 15, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.34037172F+03
ITERATION 16, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.34036401F+03
ITERATION 17, LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = 0.34036400F+03

```

MAXIMUM-LIKELIHOOD ANALYSIS OF TYPES  
ARTIFICIAL CLUSTERS METHOD-1 1 PRELIMINARY ITERATIONS

SAMPLE SIZE = 222  
NUMBER OF VARIABLES = 2  
NUMBER OF TYPES = 3

ITERATION NUMBER 17

LIKELIHOOD OF 3 TYPES IN THIS SAMPLE = +.34036409E+3

CHARACTERISTICS OF THE WHOLE SAMPLE

MEANS	
.46	.39
STANDARD DEVIATIONS	
1.38	1.47
CORRELATIONS	
1.0000	.1849
.1849	1.0000

CHARACTERISTICS OF TYPE 1

THE PROPORTION OF THE POPULATION FROM THIS TYPE = .484

MEANS	
1.19	.93
STANDARD DEVIATIONS	
1.14	.91

Best Available Copy

CORRELATIONS

1.0000	.5231
.5231	1.0000

CHARACTERISTICS OF TYPE 2

THE PROPORTION OF THE POPULATION FROM THIS TYPE = .346

MEANS

.20	-1.31
-----	-------

STANDARD DEVIATIONS

1.20	.50
------	-----

CORRELATIONS

1.0000	.2462
.2462	1.0000

CHARACTERISTICS OF TYPE 3

THE PROPORTION OF THE POPULATION FROM THIS TYPE = .170

MEANS

-1.11	1.79
-------	------

STANDARD DEVIATIONS

.03	.12
-----	-----

CORRELATIONS

1.0000	.7100
.7100	1.0000

PROBABILITIES OF TYPE MEMBERSHIP

	1	2	3	4
1	.41	.58	.01	
2	.21	.71	.07	
3	.73	.26	.01	
4	.85	.14	.01	
5	.20	.79	.01	
6	.23	.77	.01	
7	.22	.77	.01	
8	.25	.75	.01	
9	.31	.69	.01	
10	.29	.71	.01	
11	.23	.76	.01	
12	.19	.80	.01	
13	.04	.96	.01	
14	.12	.87	.01	
15	.11	.87	.01	
16	.22	.78	.01	
17	.03	.95	.01	
18	.14	.86	.01	
19	.23	.77	.01	
20	.13	.87	.01	
21	.27	.72	.01	
22	.24	.75	.01	
23	.18	.82	.01	
24	.26	.74	.01	
25	.13	.88	.01	
26	.21	.79	.01	
27	.20	.78	.01	
28	.21	.77	.01	
29	.34	.65	.01	
30	.15	.86	.01	
31	.014	.984	.001	
32	.007	.993	.001	
33	.155	.844	.15	
34	.21	.974	.001	
35	.15	.995	.001	
36	.087	.913	.001	
37	.072	.978	.001	
38	.011	.999	.001	
39	.494	.514	.001	
40	.071	.93	.001	
41	.024	.974	.001	
42	.111	.883	.001	
43	.041	.953	.001	
44	.0714	.144	.001	
45	.084	.914	.001	
46	.064	.937	.001	
47	.137	.961	.001	
48	.018	.987	.001	
49	.030	.942	.001	
50	.07	.933	.001	
51	.113	.98	.001	
52	.0113	.987	.001	
53	.0142	.954	.001	
54	.028	.972	.001	
55	.037	.951	.001	
56	.069	.932	.001	
57	.072	.927	.001	
58	.151	.847	.001	
59	.292	.71	.01	
60	.114	.986	.001	

61	.041	.959	.000
62	.004	.995	.000
63	.034	.964	.000
64	.342	.626	.131
65	.017	.983	.000
66	.014	.984	.000
67	.137	.863	.000
68	.744	.254	.000
69	.081	.919	.000
70	.217	.783	.000
71	.007	.993	.000
72	.014	.986	.000
73	.004	.996	.000
74	.144	.855	.000
75	.015	.984	.000
76	.184	.000	.814
77	.020	.000	.980
78	.008	.000	.994
79	.992	.000	.008
80	.042	.000	.958
81	.001	.000	.999
82	.023	.016	.061
83	.017	.001	.982
84	.000	.000	1.000
85	.000	.000	.985
86	.015	.000	.911
87	.000	.000	.073
88	.927	.000	.000
89	.009	.000	.991
90	.004	.000	.905
91	.004	.000	.994
92	.904	.066	.028
93	.001	.000	.999
94	.000	.000	1.000
95	.003	.000	.997
96	.034	.001	.944
97	.002	.000	.998
98	.788	.000	.212
99	.963	.001	.036
100	.000	.000	1.000
101	.008	.051	.941
102	.014	.000	.984
103	.014	.000	.986
104	.004	.000	.994
105	.960	.000	.040
106	.002	.000	.998
107	.001	.000	.999
108	.919	.073	.000
109	.000	.000	1.000
110	.010	.000	.190
111	.160	.000	.031
112	.970	.000	.021
113	.477	.000	.923
114	.995	.000	.005
115	.034	.000	.964
116	.594	.000	.406
117	.991	.000	.009
118	.004	.000	.996
119	.984	.000	.016
120	.000	.000	.991
121	.160	.000	.940
122	.007	.000	.993
123	.100	.000	.994
124	.000	.000	1.000
125	.011	.000	.989
126	.000	.200	.000

127	1.000	.000	+.00000
128	.999	.001	-.000
129	1.000	.000	+.000
130	.997	.003	-.006
131	1.000	.000	+.000
132	.997	.002	-.003
133	1.000	.001	+.000
134	.993	.007	-.000
135	1.000	.000	+.000
136	.993	.000	-.007
137	.341	.658	.696
138	.477	.523	.100
139	1.000	.000	+.000
140	1.000	.000	+.000
141	1.000	.000	+.000
142	.999	.001	-.000
143	.983	.017	.117
144	1.000	.000	+.000
145	1.000	.000	+.000
146	1.000	.000	+.000
147	.987	.013	-.000
148	.994	.005	-.001
149	.997	.002	-.003
150	1.000	.000	+.000
151	1.000	.000	+.000
152	.104	.895	.000
153	1.000	.000	+.000
154	.993	.007	-.000
155	.575	.425	.000
156	.990	.001	-.000
157	.984	.016	-.000
158	.990	.001	-.000
159	1.000	.000	+.000
160	1.000	.000	+.000
161	1.000	.000	+.000
162	1.000	.000	+.000
163	1.000	.000	+.000
164	1.000	.000	+.000
165	1.000	.000	+.000
166	.992	.008	-.000
167	.997	.002	-.003
168	1.000	.000	+.000
169	.981	.019	-.019
170	.634	.362	.000
171	1.000	.000	+.000
172	.405	.595	.000
173	.997	.003	-.000
174	.999	.007	-.010
175	1.000	.000	+.000
176	1.000	.000	+.000
177	.988	.012	-.020
178	.993	.007	-.006
179	1.000	.000	+.000
180	.999	.001	-.000
181	.468	.532	.000
182	.194	.805	-.001
183	1.000	.000	+.000
184	1.000	.000	+.000
185	.451	.549	-.000
186	1.000	.000	+.000
187	1.000	.000	+.000
188	1.000	.000	+.000
189	.975	.025	-.000
190	1.000	.000	+.000
191	1.000	.000	+.000
192	1.000	.000	+.000

193	.784	.214	.000
194	1.000	.000	.000
195	.897	.000	.103
196	1.000	.000	.000
197	.151	.849	.000
198	1.000	.000	.000
199	1.000	.000	.000
200	1.000	.000	.000
201	1.000	.000	.000
202	1.000	.000	.000
203	1.000	.000	.000
204	.874	.123	.001
205	.970	.021	.000
206	1.000	.000	.100
207	1.000	.000	.000
208	.985	.015	.000
209	1.000	.000	.000
210	.995	.005	.000
211	1.000	.000	.000
212	1.000	.000	.000
213	.135	.865	.000
214	.700	.291	.000
215	1.000	.000	.000
216	.041	.959	.000
217	1.000	.000	.000
218	.506	.494	.000
219	.996	.001	.000
220	.982	.018	.000
221	.833	.167	.000
222	.999	.005	.000
223	.538	.462	.000
224	1.000	.000	.000
225	.007	.993	.000

Best Available Copy

**VII Fortran Program Listing**

```

PROGRAM TYPE0000
DIMENSION V(5,70),L(7,6),PFR(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5)TYPE0001
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1)TYPE0002
P26),HV(126),RECORD(12),SD(5),XCAP(5,10),MAT(100,100)TYPE0003
DIMENSION FMT(12)TYPE0004
COMMON X,7,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,H,HV,RECORD,SDTYPE0005
1,MX,MX1,IR,XIR,IR1,IJMNX,KLX,NX,XNX,JATMX,CONV,ITERM,LX,XRAM,MAT,ETYPE0006
2,PH,IX,IL,ITER,PROB,METH,TDUMPTYPE0007
EQUIVALENCE (XRAM,V),(A,MAT,X)TYPE0008
1 READ INPUT TAPE 3,1 I,(RFCORD(I),I=1,12),MX,NX,IRH,ITERM,CONV,IRUNTYPE0009
1,MOTH,TDUMPTYPE0010
101 FORMAT(12AA/14,1B,2I4,F8.0,314)TYPE0011
IF(MX)23,23,2TYPE0012
C MX=NUMBER OF VARIABLES. TYPE0013
C THE PROGRAM READS BATCHES OF DATA UNTIL 2' BLANK CARDS ARE ENCOUNTEREDTYPE0014
C 1HED,TYPE0015
2 IF(ITERM)3,3,4TYPE0016
C ITERM=MAXIMUM NUMBER OF PERMISSABLE ITERATIONS. TYPE0017
3 ITERM=5TYPE0018
4 IF(CONV)5,5,6TYPE0019
C CONV=CRITERION OF CONVERGENCE WHICH ALL PARAMETERS MUST SATISFY WHETHERTYPE0020
C 1WEEN SUCCESSIVE ITERATIONS. TYPE0021
5 CONV=0.0001TYPE0022
6 MX1=MX+1TYPE0023
REWIND 4TYPE0024
NNX=NXTYPE0025
NEVK=INTF(NNN/100.)+1TYPE0026
IJMNX=IMX1=(MX1+1)*(MX1+2)*(MX1+3)/24TYPE0027
KRMN=0TYPE0028
LX=0TYPE0029
READ INPUT TAPE 3,102,(FMT(I),I=1,12)TYPE0030
102 FORMAT(12AA)TYPE0031
7 CALL BLOCK(NE,3000,KRMN,KST,KEND,LONG)TYPE0032
DO 8 J=1,LANGTYPE0033
8 READ INPUT TAPE 3,FMT,(X(I),I=1,MX)TYPE0034
DO 9 J=1,LANG,NEVTYPE0035
LX=LX+1TYPE0036
DO 9 I=1,MVTYPE0037
9 XSAM (I,LX1=IX(I),J)TYPE0038
XSAM=XSAMP OF UP TO 100 CASES USED FOR INITIAL ESTIMATES OF PARAMTYPE0039
C 1ETERS. TYPE0040
WRITE TAPE 4,(IX(I),J),I=1,MX),J=1,LONG)TYPE0041
IF (KRMN)10,1,7TYPE0042
10 CALL INITIALTYPE0043
IR=IRHTYPE0044
IR=NUMBER OF TYPES ASSUMED. IF IR>0, 6 DIFFERENT ANALYSES ARE TYPE0045
C DONE ASSUMING 1 TO 6 TYPES. TYPE0046
IF (IR)11,11,12TYPE0047
11 IR=IR+1TYPE0048
12 IR1=IR+1TYPE0049
METH=MOTHTYPE0050

```

```

XIR=IR
KLX=(IR1*(IR1+1))/2
JATMX=(MY1*(MX1+1)*19)/2
C PERS(K)=PROPORTION OF POPULATION OF TYPE K.
DO 913 K=2,IR1
IF(12M)13,13,912
912 IF(PERS(K))13,13,913
13 PERS(K)=1./XIR
913 CONTINUE
ITER=0
IDIV=0
PROBA=-104A576.0005
14 ITER=ITER+1
IF(DETERM(1)) 3500,3501,3501
3500 DETERM(1)=0.0
GO TO 346
3901 CALL MOMENT
WRITE OUTPUT TAPE 2,9916,ITER,IR,PROB
9916 FORMAT(10W ITERATION 13,15W LIKELIHOOD OF 13,23W TYPES IN THIS S
ATMPLF RE18,A)
IF(PHOB=PROBA)346,347,347
346 IDIV=IDIV+1
ITER=ITER+1
WRITE OUTPUT TAPE 2,9919,ITERA
9915 FORMAT(10W ITERATION 13,9W DIVERGES)
DO 3445 K=1,JATMX
3445 B(K)= 1.5*B(K)
IF(.1DIV-1)115,3446,115
3446 DO 3447 K=1,JATMX
3447 B(K)= -B(K)
GO TO 115
347 IDIV=0
IF(METH=)344,348,348
344 IF(.ITER=IDIMP) 348,348,3444
3444 METH=METH
C IDUMPS THE NUMBER OF PRELIMINARY ITERATIONS BY SUCCESSIVE
C SUBSTITUTIONS BEFORE NEWTON-RAPHSON ITERATION FOR METH=1,-2,OR -3
140 PROBA=PROB
IF(SENSE SWITCH 3)349,249
349 WRITE OUTPUT TAPE 2,9910
9920 FORMAT(7W MOMENT)
WRITE OUTPUT TAPE 2,9914,((V(I,J),JB1,KLX),1=1,IJMNX)
WRITE OUTPUT TAPE 2,9914,(((COV(I,J,K),1=1,MN),JB1,MN),KB1,IR1)
WRITE OUTPUT TAPE 2,9914,(((COV(IN(I,J,K)),1=1,MN),JB1,MN),KB1,IR1)
249 CONTINUE
114 CALL NEUTON
IF(SENSE SWITCH 3) 390,250
150 WRITE OUTPUT TAPE 2,9913
9911 FORMAT(7W NEUTON)
WRITE OUTPUT TAPE 2,9914,((AC(I,J),1=1,JATMX),JB1,JATMX)

```

```

        WRITE OUTPUT TAPE 2,9914,(H(I),I=1,JATMX)
9914 FORMAT(6F12.6)          TYPE 0103
250 CONTINUE                  TYPE 0102
      IF(METH) 115,115,248      TYPE 0103
C       IF METH = 0, SUCCESSIVE SUBSTITUTION IS USED, OTHERWISE
C       NEWTON-RAPSON ITERATION.
240 CALL MATINV(A,JATMX,B,1,DA)  TYPE 0104
      IF(SENSE SWITCH3) 351,251  TYPE 0105
351 WRITE OUTPUT TAPE 2,9912          TYPE 0106
9912 FORMAT(7M MATINV)
      WRITE OUTPUT TAPE 2,9914,((A(I,J),I=1,JATMX),J=1,JATMX)
      WRITE OUTPUT TAPE 2,9914,(B(I),I=1,JATMX)          TYPE 0107
251 CONTINUE                  TYPE C108
      115 CALL RAPHSAN          TYPE 0109
      IF(IRUN) 15,16,15          TYPE 0110
C       IF IRUN=0, ONLY THE LAST ITERATION IS PRINTED.
15 CALL RESULT                  TYPE 0111
16 IF(ITER=ITERM)17,19,19      TYPE 0112
17 DO 18 K=1,JATMX            TYPE 0113
      TA=B(K)
      TABABSF(TA)
      IF(TA=CONV)18,18,14      TYPE 0114
18 CONTINUE                  TYPE 0115
19 IF(IRUN) 21,20,21          TYPE 0116
20 CALL RESULT                  TYPE 0117
21 CALL PLACE                  TYPE 0118
      IF(IRN)22,22,1          TYPE 0119
22 IF(IR=0)11,1,1          TYPE 0120
23 END FILE 2                  TYPE 0121
      CALL EXIT                  TYPE 0122
      END(0,1,0,0,0)          TYPE 0123

```

```

SUBROUTINE INITIAL                                         TYPE 0132
  DIMENSION V(5,3010),Z(6),PER(7),PERB(7),AV(5,7),COV(5,5,7),COVIN(5) TYPE 0133
  1,5,7),ALPHA(7),DETERM(7),G(7),TX(4),ID(4),V(126,26),A(126,126),B(1) TYPE 0134
  226),BV(126),RECORD(12),SR(5),XSAM(5,100),MAT(1,0,100)      TYPE 0135
  COMMON X,Z,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,B,RV,RECORD,SD TYPE 0136
  1,MX,MX1,IR,XIR,IR1,IJMNX,LX,NX,XNX,JATMX,CONV,ITERM,LX,XSAM,MAT,E TYPE 0137
  2PH,IX,ID,ITER,PROR,METH,TDUMP                                TYPE 0138
  EQUIVALENCE (XSAM,V),(A,MAT,X)                                 TYPE 0139
C   THIS SUBROUTINE DETERMINES INITIAL ESTIMATES OF THE MEANS AND COVARIANCE TYPE 0140
C   ARANCES OF THE TYPES BY APPLYING A CLUSTERING PROCEDURE TO A SAMPLE TYPE 0141
C   PE OF UP TO 100 CASES.                                         TYPE 0142
C   FIRST THE SAMPLE MEANS AND COVARIANCES ARE DETERMINED.        TYPE 0143
  XLX=0
  DO 22 I=1,MX                                              TYPE 0144
  AV(I,1)=0.
  DO 1 K=1,NV                                              TYPE 0145
  1 AV(I,1)=AV(I,1)+XSAM(I,K)                                TYPE 0146
  22 AV(I,1)=AV(I,1)/XLX                                     TYPE 0147
  DO 3 J=1,MY                                              TYPE 0148
  DO 3 Je=1,MY                                              TYPE 0149
  COV(I,J,1)=0.
  DO 2 K=1,LX                                              TYPE 0150
  2 COV(I,J,1)=COV(I,J,1)+XSAM(I,K)*XSAM(J,K)               TYPE 0151
  COV(I,J,1)=COV(I,J,1)/XLX-AV(I,1)*AV(J,1)                 TYPE 0152
  3 COV(I,J,1)=COV(I,J,1)                                     TYPE 0153
  DO 4 L=1,MY                                              TYPE 0154
C   AN LX BY LX MATRIX IS COMPUTED. AN ELEMENT CORRESPONDING TO A PAIR TYPE 0155
C   1 OF POINTS IS 1 IF AND ONLY IF BOTH POINTS LIE WITHIN A BOX WHOSE TYPE 0156
C   2 SIDES ARE ONE STANDARD DEVIATION LONG.                   TYPE 0157
  SDV=COV(I,I,1)                                             TYPE 0158
  4 SD(I)=SQRTF(SDV)                                         TYPE 0159
  DO 7 K=1,LX                                              TYPE 0160
  DO 7 L=K,LX                                              TYPE 0161
  DO 5 I=1,MY                                              TYPE 0162
  R=XSAM(I,K)-XSAM(I,L)                                     TYPE 0163
  R=ARSF(R)                                                 TYPE 0164
  IF(R>SD(I))5,6,6                                         TYPE 0165
  5 CONTINUE                                                 TYPE 0166
  MAT(L,K)=01                                         TYPE 0167
  MAT(K,L)=01                                         TYPE 0168
  GO TO 7                                                 TYPE 0169
  6 MAT(L,K)=00                                         TYPE 0170
  MAT(K,L)=00                                         TYPE 0171
  7 CONTINUE                                                 TYPE 0172
  DO 14 M=2,7                                              TYPE 0173
C   THE CENTROID OF A CLUSTER IS THE POINT WITH THE GREATEST NUMBER OF TYPE 0174
C   OTHER POINTS IN A BOX AROUND IT.                           TYPE 0175
C   FIRST THE INDEX OF THE POINT OF GREATEST DENSITY IS FOUND. THE TYPE 0176
C   MEANS ARE THE CO-ORDINATES OF THE DENSEST POINT.          TYPE 0177
  MAXOL=0
  DO 11 K=1,NV
  L=0
  DO 9 R=L+1,LX

```

```

      A LB$1 H=MAT(1,1)
      IF(I H-MAX)11,12,9
      9 MAX=LR
      LAKR
10 CONTINUE
      DO 11 I=1,MX
      AV(I,M)=X$AM(I,LAK)
      DO 11 J=1,MY
      COV(I,J,M)=COV(I,J,1,1)
11 COV(I,J,M)=COV(I,J,1,M)
      DO 14 K=1,LX
      IF(MAT(K,I,AK))14,14,12
12 DO 13 L=1,IX
13 MAT(K,L)=0,
      ALL PREVIOUSLY CLUSTERED POINTS ARE ERASSED. THE CENTROID OF THE
C      NEXT CLUSTER THUS WILL BE THE POINT WITH THE GREATEST NUMBER OF PREV
C     IOUSLY UNCLUSTERED POINTS WITHIN A BOX AROUND IT.
14 CONTINUE
      DO 15 I=1,MX
      DO 15 J=1,MY
15 AV(I,J)=COV(I,J,1)
      CALL MATINV(A,MX,R,0,DA)
      AD=A$MF(DA)/RA
      DA=(SQRTF(AD/DA))=AD
      DO 16 K=1,7
      PERB(K)=1.0
      DETERM(K)=DA
      DO 16 I=1,MX
      DO 16 J=1,MY
16 COV(I,J,K)=AV(I,J)
      THE ROUTINE ALSO READS IN INITIAL ESTIMATES UNTIL A BLANK CARD
C      IS ENCOUNTERED.
17 READ INPUT TAPE 3,122,K,RA
122 FORMAT(14,F4.4)
      IF(K) 18,21,18
18 KOK=1
      PERB(K)=DA
      READ INPUT TAPE 3,123,(AV(I,K),I=1,MX)
      READ INPUT TAPE 3,123,(RD(I),I=1,MY)
192 FORMAT(9FA.4)
      DO 19 J=1,MY
      READ INPUT TAPE 3,123,(COV(I,J,K)=RD(I)=RD(J))
      DO 19 I=1,MX
      COV(I,J,K)=COV(I,J,K)+RD(I)=RD(J)
19 AV(I,J)=COV(I,J,K)
      CALL MATINV(A,MX,R,0,DA)
      AD=A$MF(DA)/RA
      DA=(SQRTF(AD/DA))=AD
      DETERM(K)=DA
      DO 20 I=1,MX
      DO 20 J=1,MY
20 COV(I,J)=AV(I,J)
      GO TO 17
21 RETURN
      END(,,1,0,n,1)

```

```

SUBROUTINE DENSITY          TYPE 0240
DIMENSION Y(5,30),Z(5),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5) TYPE 0241
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1) TYPE 0242
226),BV(126),RECORD(12),SD(5),XSAM(5,100),MAT(100,100) TYPE 0243
COMMON X,Z,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,B,RV,RECORD,SD TYPE 0244
1,MX,MX1,IR,XIR,IR1,IJMN,KLX,NX,XNX,JATHX,CONV,ITERM,LX,XSAM,MAT,E TYPE 0245
2PH,IX,ID,ITER,PROR,METH,LDUMP TYPE 0246
EQUIVALENCE (XSAM,V),(A,MAT,Y) TYPE 0247
C FOR CURRENT ESTIMATES OF THE PARAMETERS, THIS ROUTINE COMPUTES- TYPE 0248
C ALPHA(K)=MULTIVARIATE NORMAL DENSITY FOR TYPE K AT POINT Z. TYPE 0249
C EPH=DENSITY AT Z FOR THE MIXTURE OF DISTRIBUTIONS TYPE 0250
C G(K)=ALPHA/EPH AT POINT Z. TYPE 0251
C DO 3 K=2,IR1 TYPE 0252
C AL=0. TYPE 0253
C DO 2 I=1,MY TYPE 0254
C IA=I+1 TYPE 0255
C DO 1 J=1,MY TYPE 0256
1 AL=AL-COVIN(J,I,K)*(Z((IA)-AV(I,K)))*(Z((J+1)-AV(J,K))) TYPE 0257
2 AL=AL+0.5*COVIN(I,(I,K)*(Z((IA)-AV(I,K)))*#2 TYPE 0258
3 ALPHA (K)= DETERM (K)*EXP(AL) TYPE 0259
EPH=0. TYPE 0260
DO 4 K=2,IR1 TYPE 0261
4 EPH=EPH+PERS(K)*ALPHA(K) TYPE 0262
DO 5 K=2,IR1 TYPE 0263
5 G(K)=ALPHA(K)/EPH TYPE 0264
G(1)=1. TYPE 0265
C THE ADDITION OF A DUMMY TYPE OF DENSITY GRI SIMPLIFIES THE COMPUTATION TYPE 0266
C 1TION OF LOWER ORDER MOMENTS WITH ONE OR MORE TYPES OMITTED. TYPE 0267
C RETURN TYPE 0268
C END(L=1,C,~,0) TYPE 0269

```

```

SUBROUTINE MOMENT                                     TYPE0270
DIMENSION X(5,310),7(6),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE0271
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1TYPE0272
226),HV(126),RECORD(12),SR(5),XSAM(5,100),MAT(100,100)           TYPE0273
COMMON X,7,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,B,HV,RECORD,SDTYPE0274
1,MX,MX1,IR,XIR,IR1,IJMNX,KLX,NX,XNX,JATMX,CONV,ITERM,LX,XSAM,MAT,ETYPE0275
2PH,IX,ID,ITER,PROB,METH,TDUMP                           TYPE0276
EQUIVALENCE (XSAM,V),(A,MAT,X)                         TYPE0277
C THIS ROUTINE COMPUTES THE GENERALIZED MOMENTS VI(IJMN,KL) FOR VARIATYPE0278
C TABLES I,J,M, AND N, AND TYPES K AND L. THE MOMENTS REALLY HAVE SIX TYPE0279
C 2INDICES BUT ARE STORED AS A 2-DIMENSIONAL MATRIX. TRIANGULAR INDEXTYPE0280
C SING IS USED TO ELIMINATE DUPLICATION.                   TYPE0281
DO 1 IJMN=1,IJMNX                                         TYPE0282
DO 1 KL=1,KLX                                           TYPE0283
1 V(IJMN,KL)=0.                                         TYPE0284
KRMN=0                                                 TYPE0285
C INITIALIZE                                         TYPE0286
REWIND 4                                              TYPE0287
PROB$XNX                                         TYPE0288
DO 10 K=2,1R1                                         TYPE0289
10 PROB$PROB=XNX=PERS(K)                               TYPE0290
2 CALL BLOCK(NX,3000,KRMN,KST,KEND,LONG)             TYPE0291
READ TAPE 4,1(X(I,J),I=1,MX),J=1,LONG)              TYPE0292
C TAPE 4 IS READ IN BLOCKS OF 3010 CASES.            TYPE0293
DO 17 KA=1,LONG                                         TYPE0294
Z(1)=1.0                                              TYPE0295
C THE ADDITION OF A DUMMY VARIABLE SIMPLIFIES THE COMPUTATION OF LTYPE0296
C LOWER ORDER MOMENTS OMITTING ONE OR MORE VARIABLES.   TYPE0297
DO 3 I=1,MV                                         TYPE0298
3 Z(I+1)=X(I,KA)                                       TYPE0299
CALL DENSITY                                         TYPE0300
C PROBABILITY OF SAMPLE                                TYPE0301
IF(EPH)7,7,8                                         TYPE0302
7 INDEX$KST$KA=1                                      TYPE0303
WRITE OUTPUT TAPE 2,120,INDEX,PPM                     TYPE0304
180 FORMAT(49H NEGATIVE PROBABILITY DENSITY FOR OBSERVATION 19,3H  * TYPE0305
1F16.9)                                               TYPE0306
GO TO 9                                              TYPE0307
8 PROB$PROB=LOGF(EPH)                                TYPE0308
9 KLN=0                                              TYPE0309
IF(METH) 12,12,11                                     TYPE0310
11 DO 4 K=1,1R1                                         TYPE0311
DO 4 L=1,K                                         TYPE0312
KL=KL+1                                             TYPE0313
GT1=G(K)*G(L)                                       TYPE0314
IJMN=0                                              TYPE0315
DO 4 I=1,MV1                                         TYPE0316
GT2=GT1*Z(I)                                       TYPE0317
DO 4 J=1,I                                         TYPE0318
GT3=GT2*Z(J)                                       TYPE0319

```

```

DO 4 M=1,J
GT4 = GT3*Z(H)
DO 4 N=1,M
IJMN=IJM+1
4 V(IJMN,KL)=V(IJMN,KL)+GT4*Z(K)
17 CONTINUE
15 IF(KHMN)5,5,2
5 DO 6 IJMN=1,IJMX
DO 6 KL=1,KLX
6 V(IJMN,KL)=V(IJMN,KL)/KMN
HFMN
12 KL=1
DO 13 K=1,TH1
FOR METHOD, WE DO NOT NEED ALL THE MOMENTS
IJMN=2
DO 14 J=2,NY1
GT1= Z(1)* G(K)
JS=1
DO 16 JS=1,I
V(IJMN,KL)= V(IJMN,KL) + -5**.7(J)
JS=JS+J
14 IJMN=IJM+1
DO 15 L=1,K
V(1,KL)=V(1,KL)+G(K)*H(L)
15 KL=KL+1
GO TO 17
ENDC ,1,1,1,1

```

TYPE0320  
TYPE0321  
TYPE0322  
TYPE0323  
TYPE0324  
TYPE0325  
TYPE0326  
TYPE0327  
TYPE0328  
TYPE0329  
TYPE0330  
TYPE0331  
TYPE0332  
TYPE0333  
TYPE0334  
TYPE0335  
TYPE0336  
TYPE0337  
TYPE0338  
TYPE0339  
TYPE0340  
TYPE0341  
TYPE0342  
TYPE0343  
TYPE0344  
TYPE0345  
TYPE0346

```

FUNCTION V(IA,JA,MA,NA,KA,LA ) TYPE0347
  DIMENSION V(5,7),U(5,7),Z(6),PFR(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE0348
  1,5,7),ALPHA(7),DFTERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1TYPE0349
  /26),HV(126),RECORD(12),SN(5),XSAM(5,100),MAT(1,1,100) TYPE0350
  COMMON X,7,PER,PERS,AV,COV,COVIN,ALPHA,DFTERM,G,V,A,H,HV,RECORD,SDTYPE0351
  1,MX,MX1,IR,XIR,IR1,IJMN,KLX,NX,YNX,JATMX,CONV,ITERM,LX,XSAM,MAT,ETYPE0352
  2PM,IX, ID,ITER,PROB,METH, IDUMP TYPE0353
  EQUIVALENCE (XSAM,V),(A,MAT,Y) TYPE0354
C THIS ROUTINE FINDS THE MOMENT CORRESPONDING TO 6 INDICES BY COMPUTTYPE0355
C ING THE 2 INDICES OF THE MOMENT IN THE MATRIX V. TYPE0356
C
  IX(1)=IA TYPE0357
  IX(2)=JA TYPE0358
  IX(3)=MA TYPE0359
  IX(4)=NA TYPE0360
C THE FIRST 4 INDICES ARE PUT IN ORDER FROM SMALLEST TO LARGEST TYPE0361
  DO 3 J=1,4 TYPE0362
  KAT=100 TYPE0363
  DO 2 I=1,4 TYPE0364
  IF(KAT>=IX(I)) 2,2,1 TYPE0365
  1 KAT=IX(I) TYPE0366
  K=1 TYPE0367
  GO TO 2 TYPE0368
  2 CONTINUE TYPE0369
  IX(K)=1 TYPE0370
  3 ID(J)=KAT TYPE0371
C THE LAST 2 INDICES ARE PUT IN ORDER FROM LARGER TO SMALLER TYPE0372
  IF (KA=LA)4,5,5 TYPE0373
  4 KAT=LA TYPE0374
  LAT=KA TYPE0375
  GO TO 6 TYPE0376
  5 KAT=KA TYPE0377
  LAT=LA TYPE0378
  6 IJMN=1 TYPE0379
C THE APPROPRIATE INDICES OF THE MOMENT MATRIX V ARE COMPUTED. TYPE0380
  DO 8 I=1,4 TYPE0381
  LPR=1 TYPE0382
  DO 7 J=1,1 TYPE0383
  7 LPR=(LPR+(ID(I)+2+J))/J TYPE0384
  8 IJMN=IJMN+1 PR TYPE0385
  KL=(KAT-(KAT-1))/2+LAT TYPE0386
  U=V(IJMN,KL) TYPE0387
  RETURN TYPE0388
  END(0,0,0,0) TYPE0389

```

```
C      FUNCTION DIFE(I,J)
C      THIS IS JUST THE KRONECKER DELTA
C      IF(I==J)1,2,1
1      UNES .
      GO TO 3
2      ONE=1.
3      HF=1/N
      END
```

TYPE n390  
TYPE n391  
TYPE n392  
TYPE n393  
TYPE n394  
TYPE 0395  
TYPE n396  
TYPE n397

```

FUNCTION F(X(1),I,J,A,IS,P) TYPE n398
DIMENSION X(7,3),I(7,6),P(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5) TYPE n399
,,5,7),ALPHA(7),DETERM(7),G(7),IX(4),IP(4),U(12A,2B),A(126,126),B(17) TYPE n400
226),HV(12A),RECORD(12),SH(5),XSAM(5,12),WATE(1,1,1) TYPE n401
COMMON X,2,PFH,PERS,AV,COV,COVTN,ALPHA,DETERM,G,V,A,H,HV,RECORD,SDTYPE n402
1,MX,NX2,IP,XIH,[4],SUMX,KLX,NY,XXX,ATMX,CONV,ITERMLX,XSAM,WATE,ETYPE n403
2PH,IX,TD,ITER,PROR,METH,TRIMP TYPE n404
EQUIVALENCE (XSAM,V),(A,WATE,X) TYPE n405
C EM IS A TERM WHICH APPEARS IN THE PARTIAL DERIVATIVE OF THE MAXIMUMTYPE n406
C 1N LIKELIHOOD EQUATIONS WITH RESPECT TO THE TYPE MEANS. TYPE n407
C EM= TYPE n408
C IAI=1 TYPE n409
C DO 1 I=1,MV TYPE n410
C 1B1 TYPE n411
1 EM=EM+COVIN(I,A,1,IP)*(U(1+1,I)+U(1+1,IP+1))-AV(IH,IP)*U(1+1,IP) TYPE n412
1J,IP,IS) TYPE n413
RETURN TYPE n414
END(L,1,..,..) TYPE n415

```



```

FUNCTION DERV(IU,IS,IA,IR,IP) TYPE n434
  UTIME,SIG1,VIN,TR,1,2(6),PERS(7),PERP(7),AV(5,7),COV(4,5,7),COVIN(S)TYPE n435
  5,6,7),ALPHA(7),BETAP(7),G(1),IX(4),ID(4),V(12A,OH),A(120,126),R(1)TYPE n436
  22A),AV(12A),RECORD(12),SP(5),XSAM(5,1P),XMAT(1,1) TYPE n437
  COMMN,X(7,PEN,PERS,AV,COV,G,VIN,ALPHA,BETAP,H,G,V,A,M,H,V,RECORD,SD)TYPE n438
  1,MY,FX1,IR,XIR,IH1,LUNNK,KLX,NY,LYNX,DATNX,DU,V,ITEM,DLX,XSAM,XMAT,ET)TYPE n439
  2PH,TV,TD,ITER,PROB,METH,TRIML TYPE n440
  EQUIVALENCE (XSAM,V),(AMAT,Y) TYPE n441
  THIS ROUTINE COMPUTES THE DERIVATIVE OF THE IUS MOMENT WITH TYPE n442
  RESPECT TO THE AND PARAMETER. TYPE n443
  IF(IA-1)1,2,2 TYPE n444
1 DERV = -UTM(IU,IS,IA,IR,IP) TYPE n445
  GO TO 1 TYPE n446
2 IF(IR-1)3,4,6 TYPE n447
3 DERV=PERS(IP)=EM(IU,IS,IA,IR,IP) TYPE n448
  IF(IR-1)5,6,6 TYPE n449
4 DERV=DERV+EM(IU,IS,IA,IR,IP) TYPE n450
  GO TO 1 TYPE n451
5 DERV=PERS(IP)=SIG(IU,IS,IA,IR,IP) TYPE n452
  IF(IR-1)6,7,8 TYPE n453
7 DERV=DERV+SIG(IU,IS,IA,IR,IP) TYPE n454
8 IF((IA-10)*2,9,10) TYPE n455
9 DERV=0.5*DERV TYPE n456
10 RETURN TYPE n457
END TYPE n458

```

```

SUBROUTINE NEWTON          TYPE 0459
DIMENSION V(5,7),A(7),PFR(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE 0460
1,5,7),ALPHA(7),DETERM(7),G(7),IX(4),IN(4),V(126,28),A(126,126),U(1TYPE 0461
226),HV(126),REFORM(12),SM(5),XRAM(5,100),MAT(100,100)           TYPE 0462
COMMON X,Z,PFH,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,H,HV,RECORD,SDTYPE 0463
1,NY,MX1,IR,XIR,IR+,IJMX,LX,NX,XNN,JATMX,CONV,ITER,LX,XRAM,MAT,ETYPE 0464
2PM,IX,ID,ITER,PROB,METH,ITERMP
EQUIVALENCE IXRAM,V,(4,4AT,Y)                                     TYPE 0465
C THIS ROUTINE SETS UP THE NEWTON-RAPHSON MATRIX EQUATION FOR THE CTYPE 0466
C JANGES IN THE PARAMETERS FOR THE NEXT ITERATION.                   TYPE 0467
C METH=+1 OR -1 IMPLIES SIMPLIFIED MAXIMUM-LIKELIHOOD EQUATIONS   TYPE 0468
C METH=+2 OR -2 IMPLIES COMPLETE MAXIMUM-LIKELIHOOD EQUATIONS     TYPE 0469
C METH=+3 OR -3 IMPLIES COMPLETE EQUATIONS DIVIDED BY UOS           TYPE 0470
C IF(METH)42,42,8          TYPE 0471
A IAT=
DO 27 K=2,IR1
UJSU(1,1,1,1,K,1)                                         TYPE 0472
DO 27 J=1,MX1
UJSU(1,1,1,1,J,MX1)                                         TYPE 0473
DO 27 I=J,MX1
UJSU(1,1,1,1,I,MX1)                                         TYPE 0474
JAT=
IAT=IAT+1
DO 27 L=2,IR1
DO 27 N=1,MX1
DO 27 M=N,MX1
JAT=JAT+1
TEMP=DFRV(I,J,K,M,N,L)
IF(I=1) 27,27,9
9 GO TO (12,1,-1),MFK
10 TEMP=TEMP+AV(I,-1,K)*DFRV(J,-1,K,M,N,L)                TYPE 0475
11 IF(J=1) 12,12,11
12 TEMP=TEMP+(COV(I,-1,J,-1,K)-AV(I,-1,K)*AV(J,-1,K))*DFRV(I,-1,K,M,N,L) TYPE 0476
13 1-AV(J,-1,K)*DFRV(I,-1,K,M,N,L)                         TYPE 0477
14 IF(K=L) 27,13,27
15 IF(N=1) 14,14,17
16 GO TO (16,15,14),MFK
17 TEMP=TEMP+ONE(I,M)*UJSU(I,-1,M)+ONE(J,M)*UJSU(J,-1,M)-UOS* TYPE 0478
18 ONE(I,M)*AV(I,-1,K)+ONE(J,M)*AV(J,-1,K)=AV(I,-1,K))           TYPE 0479
19 GO TO 27
20 TEMP=TEMP-ONE(J,-1)*ONE(I,M)+(1,- ONE(I,-1))*(ONE(I,M)*AV(I-1,K)+ TYPE 0480
1ONE(J,M)*AV(I-1,K))
21 GO TO 27
22 IF(J=1) 27,27,10
23 TEMP=COV(I,-1,M-1,L)*COV(J-1,N-1,L)+COV(I-1,N-1,L)*COV(J-1,M-1,L) TYPE 0481
24 GO TO (2,1,9,10),MFK
25 TEMP=TEMP+TFN
26 TEMP=TEMP+TFN
27 A(IAT,JAT)=TEMP
L8
DO 33 K=2,IR1
L=L+1

```

```

U0S=U(1,1,1,1,K,1)                                TYPE:0914
B(L)=1,0,0NS                                     TYPE:0915
DO 29 I=2,MX1                                     TYPE:0916
L=L+1
B(L)=AV(I-1,K)                                    TYPE:0917
GO TO (29,28,29),MFK                            TYPE:0918
28 B(L)=B(L)=0,0NS                               TYPE:0920
29 B(L)=B(L)=I(I-1,1,1,I+K,1)                     TYPE:0921
DO 33 J=2,MX1                                     TYPE:0922
DO 33 I=J,MX1                                     TYPE:0923
L=L+1
GO TO (31,31,31),MFK                            TYPE:0924
30 B(L)=U0S=(COV(I-1,J-1,K)-AV(I-1,K)*AV(J-1,K))+AV(I-1,K)*
     U(I-1,J,K,1)*AV(J-1,K)+U(I-1,1+I,K,1)      TYPE:0925
     GO TO 33                                         TYPE:0927
31 B(L)=COV(I-1,J-1,K)*AV(I-1,K)*AV(J-1,K)      TYPE:0928
32 B(L)=B(L)=I(I-1,I,J,K,1)                      TYPE:0930
     GO TO (37,37,34),MFK                           TYPE:0931
34 IAT=0                                         TYPE:0932
DO 36 K=2,IR1                                     TYPE:0933
JAT=0
DO 35 L=2,IR1                                     TYPE:0934
DO 35 N=1,MX1                                     TYPE:0935
DO 35 MN=MN,MX1                                   TYPE:0936
JAT=JAT+1                                         TYPE:0937
35 BV(JAT)= DPMV(I-1,K,M,N,L)/U(I-1,1+I,K,1)   TYPE:0938
DO 36 J=1,MX1                                     TYPE:0940
DO 36 I=J,MX1                                     TYPE:0941
IAT=IAT+1                                         TYPE:0942
JAT=0
DO 36 L=2,IR1                                     TYPE:0943
DO 36 N=1,MX1                                     TYPE:0945
DO 36 MN=MN,MX1                                   TYPE:0946
JAT=JAT+1                                         TYPE:0947
36 A(IAT,JAT)= A(IAT,JAT)+B(IAT)*BV(JAT)       TYPE:0948
37 RETURN                                         TYPE:0949
42 DO 44 I=1,IR
     DO 43 J=1,I
     A(I,J)=U(I-1,1+I,I+1+J-1,J+1)               TYPE:0950
43 A(J,I)=A(I,J)                                 TYPE:0951
44 B(I)= U(I-1,1+I,I+1,I+1)-1.0                 TYPE:0952
CALL MATINV(A,IR,0,1,0A)
L=IR
DO 42 K=2,IR1                                     TYPE:0953
PBM= U(I-1,1+I,I+1,K)                           TYPE:0954
DO 42 I=1,MX
L=L+1
B(L)=U(I-1,1+I,I+1,K)/PBM -AV(I,K)             TYPE:0955
BV(I)= AV(I,K)*B(L)
DO 42 J=1,I
L=L+1
52 B(L)=U(I-1,J+1,I+1,K)/PBM -BV(I)*AV(J)*COV(I,J,K)  TYPE:0956
GO TO 37                                         TYPE:0957
END

```

```

SUBROUTINE RAPHSON          TYPE 0960
DIMENSION V(5,30,7),Z(6),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5)TYPE 0969
1:5,7),ALPHA(7),DETERM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1)TYPE 0970
2:6),BV(126),RECORD(12),SR(5),XRAM(5,100),MAT(100,100)           TYPE 0971
COMMON X,7,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,B,MV,RECORD,SDTYPE 0972
1:MX,MX1,IR,XIR,IR1,IJMNX,KLX,NX,XNK,JATMX,CONV,ITERM,LX,XRAM,MAT,ETTYPE 0973
2:PM,IX,ID,ITER,PROB,METH,I,DUMP                                TYPE 0974
EQUIVALENCE (XRAM,V),(A,MAT,X)                                     TYPE 0975
C THIS ROUTINE ADDS THE APPROPRIATE CHANGES TO THE PARAMETERS TO OBTTYPE 0976
CAIN THEIR VALUES FOR THE NEXT ITERATION.                           TYPE 0977
C AKAT=2.0                                         TYPE 0978
C IF(METH)=30,30,23                                         TYPE 0979
C NEWTON-RAPHSON ITERATION                                     TYPE 0980
23 IAT=1                                         TYPE 0981
MX=MX(MX1=(MX1+1))/2                                         TYPE 0982
C SHORTEN INCREMENT VECTOR UNTIL ALL PERCENTAGES ARE WITHIN BOUNDSTYPE 0983
DO 10 K=2,IR1
AKAZ=PERS(K)/(2.-B(IAT))                                      TYPE 0984
IF(AKAZ)>7.7,A                                         TYPE 0985
7 AKAZ=AKAZ + 5/R(IAT)                                       TYPE 0986
8 IF(AKAZ-AKAT)<0.1,10,10                                     TYPE 0987
9 AKAT=AKAZ                                         TYPE 0988
10 IAT=IAT+MX                                         TYPE 0989
IF(AKAT=1.0)11,13,13                                         TYPE 0990
11 DO 12 K=1,IATMX                                     TYPE 0991
12 B(K)=AKAT=B(K)                                         TYPE 0992
13 IAT=0
DO 20 K=2,IR1
IAT=IAT+1                                         TYPE 0993
PERS(K)=PERS(K)+B(IAT)                                      TYPE 0994
DO 24 IR1,MV                                         TYPE 0995
IAT=IAT+1                                         TYPE 0996
24 AV(1,K)=AV(1,K)+B(IAT)                                     TYPE 0997
DO 25 J=1,MX                                         TYPE 0998
DO 251 J,MX                                         TYPE 0999
IAT=IAT+1                                         TYPE 0999
COVIN(1,J)=COVIN(1,J)+B(IAT)                                     TYPE 0999
COVIN(1,J,1)=COVIN(1,J,1)                                     TYPE 0999
A(1,J)=COVIN(1,J,1)                                         TYPE 0999
25 A(J,I)=A(I,J)
CALL MATINV(A,MX,MV,1,0,A)
AB=AMSF(DA)/DA                                         TYPE 0999
DETERM(1)=SQRT(AB*DA)                                     TYPE 0999
IF(AB)>7.7,27,28                                         TYPE 0999
27 IND=IR1-1                                         TYPE 0999
WRITE OUTPUT TAPE 2,121,1NLFY,DA                           TYPE 0999
121 FORMAT(44W DETERMINANT OF COVARIANCE INVERSE FOR TYPE 120,1NLFY,A,9)TYPE 0999
DETERM(1)=0.1                                         TYPE 0999
28 DO 29 J=1,MX                                         TYPE 0999
DO 291 J,MX                                         TYPE 0999
29 COVIN(1,J)=A(I,J)
29 RETURN                                         TYPE 0999

```

```

C   SUCCESSIVE SUBSTITUTIONS.          TYPE0620
30 L=IR                                TYPE0621
MX=1                                TYPE0622
C   SHORTEN INCREMENT VECTOR UNTIL ALL PERCENTAGES ARE WITHIN BOUND TYPE0623
DO 36 K=2,IR1                          TYPE0624
AKA7=PERS(K)/(2.+R(K-1))              TYPE0625
IF(AKA7)37,33,34                      TYPE0626
33 AKA7 = AKA7 + .5/R(K-1)            TYPE0627
34 IF(AKA7-AKAT)39,34,36              TYPE0628
35 AKAT=AKA7                           TYPE0629
36 CONTINUE                            TYPE0630
IF(AKAT<1.e)37,39,39                  TYPE0631
37 DO 38 K=1,IR                        TYPE0632
B(K)=AKAT=R(K)                         TYPE0633
38 DO 55 K=2,IR1                      TYPE0634
PERS(K)=PERS(K)+ R(K-1)                TYPE0635
DO 52 J=1,MX                           TYPE0636
L=L+1                                 TYPE0637
47 AV(I,K)=AV(I,K)+R(L)              TYPE0638
DO 52 J=1,I                           TYPE0639
L=L+1                                 TYPE0640
49 COV(I,J,K)=COV(I,J,K)+B(L)        TYPE0641
51 COV(J,I,K)=COV(J,I,K)             TYPE0642
A(I,J)=COV(I,J,K)                   TYPE0643
A(J,I)=A(I,J)                         TYPE0644
52 CONTINUE                            TYPE0645
53 CALL MATINV(A,MX,BV,N,DA)          TYPE0646
AD=ADSF(DA)/DA                       TYPE0647
DETERM(K)=(SORTF(AD/DA))+AD          TYPE0648
DO 54 I=1,MX                           TYPE0649
DO 54 J=1,MX                           TYPE0650
54 COV(I,J,K)=A(I,J)                 TYPE0651
55 CONTINUE                            TYPE0652
GO TO 29                               TYPE0653
END                                    TYPE0654

```

```

SUBROUTINE RESULT          TYPE n655
DIMENSION V(5,300),Z(6),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5TYPE 0656
1,5,7),ALPHA(7),DETEHM(7),G(7),IX(4),ID(4),V(126,28),A(126,126),B(1TYPE 0657
226),HV(126),RECORD(12),SN(5),XCAM(5,100),MAT(100,100)           TYPE 0658
COMMUN X,2,PER,PERS,AV,COV,COVIN,ALPHA,DETERM,G,V,A,H,V,RECORD,SDTYPE 0659
1,MX,MX1,IR,XIR,IR1,IJMNK,KLK,NX,XNK,JATMX,CONV,ITERMLY,XRAM,MAT,ETTYPE 0660
2PH,IX,IN,ITER,PROB,METH,TDUMP                                     TYPE 0661
EQUIVALENCE (XCAM,V),(A,MAT,X)                                     TYPE 0662
THIS ROUTINE PRINTS OUT THE PARAMETERS DESCRIBING EACH TYPE.          TYPE 0663
1 FORMAT(1H1,15X,3HMAXIMUM,L1E1,1WOOD ANALYSIS OF TYPES//4Y,12A6)  TYPE 0664
2 FORMAT(1H1,15X,3HCHARACTERISTICS OF THE WHOLE SAMPLE//3Y)        TYPE 0665
3 FORMAT(3Y//14X,23HCHARACTERISTICS OF TYPE 14//3X)                 TYPE 0666
4 FORMAT(11Y,49HTHE PROPORTION OF THE POPULATION FROM THIS TYPE 0F6,TYPE 0667
 13/3X)                                                       TYPE 0668
5 FORMAT(3X,5HMEANS/5F12.2)                                         TYPE 0669
6 FORMAT(3X/23X,19HSTANDARD DEVIATIONS /5F12.2)                   TYPE 0670
7 FORMAT(3X/23X,12HCORRELATIONS)                                     TYPE 0671
8 FORMAT (5F12.4)                                              TYPE 0672
9 FORMAT(3X//14X,13HSAMPLE SIZE =11/10X,21HNUMBER OF VARIABLES =TYPE 0673
 112/14X,17HNUMBER OF TYPES =14//25X,1AHITERATION NUMBER 13//3X,      TYPE 0674
 213HLIKELIHOOD OF 12,23H TYPES IN THIS SAMPLE =F1A.0)             TYPE 0675
DO 1 J=1,MX
J=J+1
1 AV(J)=0.0
DO 2 J=1,MX
J=J+1
DO 2 I=J,MX
I=I+1
COV(I,J,1)=0.0
COV(I,J,1)=COV(I,J,1)+AV(J)*AV(I)
2 COV(I,J,1)=COV(I,J,1)/2.0
WRITE OUTPUT TAPE 2,1,(RECORD(1),[0],1)
WRITE OUTPUT TAPE 2,0,NX,MX,IR,ITER,IR,PHON
DO 14 K=1,12
K=K+1
IF (K) 11,11,12
11 WRITE OUTPUT TAPE 2,2
GO TO 13
12 WRITE OUTPUT TAPE 2,3,K1
WRITE OUTPUT TAPE 2,4,PERS(K)
13 WRITE OUTPUT TAPE 2,5,(AV(I,K),[0],MX)
DO 14 I=1,MX
SDV=COV(I,I,K)
AD=ADSF(SDV)/SDV
SDV=(SDV/AD)-QDV11=AD
14 SDV11=SDV
DO 19 I=1,MX
DO 19 J=1,MX
A(I,J)=COV(I,J,K)/(SN(I)*SN(J))
15 A(I,J)=A(I,J)
WRITE OUTPUT TAPE 2,6,(SN(I),[0],MX)
WRITE OUTPUT TAPE 2,7
DO 19 J=1,MX
16 WRITE OUTPUT TAPE 2,8,(A(I,J),[0],MX)
RETURN
END

```

```

SUBROUTINE PLACE
DIMENSION Y(5,7),A(1,7)(6),PER(7),PERS(7),AV(5,7),COV(5,5,7),COVIN(5,TYPE 0711
1,5,7),ALPHA(/),DFTFRM(7),G(7),IX(4),ID(4),V(128,2B),A(120,120),B(1)TYPE 0712
22A),HV(12A),RECORD(12),SN(5),XSAM(5,12),MAT(1-11) TYPE 0713
COMMON X,7,PER,PERS,AV,COV,C,VIN,ALPHA,DFTFRM,G,V,A,H,LV,RECORD,SDTYPE 0714
1,MX,1X1,1P,XIR,IR,IJMX,XLX,AY,YNN,XJATMX,CONV,ITERVAL,XSAM,MAT,TYPE 0715
2PH,1X,1D,ITER,PRDR,PRDR,METH,IDLMP TYPE 0716
EQUIVALENCE (XSAM,V),(A,MAT,Y)
THIS ROUTINE PRINTS OUT THE PROBABILITIES OF TYPE MEMBERSHIP FOR ETYPEn0717
1ACH OBSERVATION TYPE 0719
1 FORMAT(1H-,9X,32HPROGRAM// TLES OF TYPE MEMBERSHIP//,Y,7I8) TYPE 0720
2 FORMAT(14.7E.4) TYPE 0721
WRITE OUTPUT TAPE 2,1,(1,I81,1H)
KBD
KRMNRP
REWIND 4
3 CALL BLOCK(IX,51,-,KRMN,KST,READ,1LONG)
READ TAPE 4, ((X(I,J),I=1,MX), J=1,LONG)
DO A K=1,1LONG
DO 4 I=1,MV
4 Z((I+1)BY(1,K))
CALL DENSITY
KRMN=0
DO 5 I=2,IR1
5 G((I)BYG((1))=PERS((1))
6 WRITE OUTPUT TAPE 2,2,KB,(G((1)),I=2,IR1)
IF(KRMN)=7,7,3
7 RETURN
ENDC-1,0,0,0,0

```

```

C      SUBROUTINE _CCKINY,LALK,KHMN,KST,KEND,LONG)          TYPE 0739
C      THIS ROUTINE COMPUTES NUMBERS USEFUL IN THE CONTROL OF THE INPUT   TYPE 0740
C      AND OUTPUT OF LISTS OF LENGTH NX IN BLOCKS OF LENGTH LALK.           TYPE 0741
C      KHMN=NUMBER OF ITEMS REMAINING IN THE LIST                      TYPE 0742
C      LONG=LENGTH OF CURRENT BLOCK                                     TYPE 0743
C      KST AND KEND ARE THE STARTING AND ENDING INDEXES FOR THE ITEMS IN TYPE 0744
C      THE CURRENT BLOCK.                                              TYPE 0745
C      IF (KHMN)2,1+2                                                 TYPE 0746
1     KHMNONX                                         TYPE 0747
      KEND=0                                         TYPE 0748
2     LONG=LALK                                         TYPE 0749
      KST=KEND+1                                         TYPE 0750
      IF (KHMN=L) X13,4,4                                TYPE 0751
3     LONG=KHMN                                         TYPE 0752
4     KEND=KEND+1 LONG                                 TYPE 0753
      KHMN=KHMN-1 LONG                                 TYPE 0754
      RETURN                                           TYPE 0755
      END                                              TYPE 0756

```



```

265 IF(M) 260, 261, 21.
21 DO 250 L=1, M
220 SWAP=B(1ROW,L)
230 B(1ROW,1)=B(1COLUMN,L)
240 B(1COLUMN,L)=SWAP
250 INDEX(1,1)=1ROW
260 INDEX(1,2)=1COLUMN
270 PIVOT(1)=A(1COLUMN,1COLUMN)
280 DETERM=DETERM*PIVOT(1)
C   DIVIDE PIVOT ROW BY PIVOT ELEMENT
330 A(1COLUMN,1COLUMN)=1.0
340 DO 350 L=1,N
350 A(1COLUMN,L)=A(1COLUMN,L)/PIVOT(1)
360 IF(M) 361, 361, 34.
360 DO 370 L=1,M
370 B(1COLUMN,L)=B(1COLUMN,L)/PIVOT(1)
C   REDUCE NON-PIVOT ROWS
380 DO 590 L=1,N
390 IF(L>1-1COLUMN) 40, 551, 41
400 TA=A(L1,1COLUMN)
410 A(L1,1COLUMN)=0.0
420 DO 430 L=1,N
430 A(L1,L)=A(L1,L)-A(L1,1COLUMN)*T
440 IF(M) 551, 451, 4A1
451 DO 540 L=1,M
540 B(L1,L)=B(L1,L)-B(L1,1COLUMN)*T
550 CONTINUE
C   INTERCHANGE COLUMNS
560 DO 710 I=1,N
570 L=N+1-I
580 IF (INDEX(I,1)=INDEX(L,2)) 610, 710, 630
590 JROW=INDEX(L,1)
600 JCOLUMN=INDEX(L,2)
610 DO 710 K=1,N
620 SWAP=A(K,JROW)
630 A(K,JROW)=A(K,JCOLUMN)
640 A(K,JCOLUMN)=SWAP
715 CONTINUE
730 CONTINUE
740 RETURN
END
ENDF,1,1,1,1,1,1

```

TYPE0798  
TYPE0799  
TYPE0800  
TYPE0801  
TYPE0802  
TYPE0803  
TYPE0804  
TYPE0805  
TYPE0806  
TYPE0807  
TYPE0808  
TYPE0809  
TYPE0810  
TYPE0811  
TYPE0812  
TYPE0813  
TYPE0814  
TYPE0815  
TYPE0816  
TYPE0817  
TYPE0818  
TYPE0819  
TYPE0820  
TYPE0821  
TYPE0822  
TYPE0823  
TYPE0824  
TYPE0825  
TYPE0826  
TYPE0827  
TYPE0828  
TYPE0829  
TYPE0830  
TYPE0831  
TYPE0832  
TYPE0833  
TYPE0834  
TYPE0835  
TYPE0836  
TYPE0837  
TYPE0838  
TYPE0839